Malware Detection using machine learning

1st Subhajit Dutta
Computer Science Department
Graphic Era Hill University
Dehradun, India
duttasubhajit88@gmial.com

2nd Sourabh Singh Rawat
Computer Science Department
Graphic Era Hill University
Dehradun, India
rawatsaura7890@gmail.com

3rd Aditya Gupta
Computer Science Department
Graphic Era Hill University
Dehradun, India
gg209404@gmail.com

***Abstract*—The burgeoning complexity and escalating volume of malware threats have necessitated the development of sophisticated and efficient malware analysis techniques. Machine learning algorithms have demonstrated considerable promise in this domain, by mechanizing the malware analysis process and identifying hitherto unknown malware samples. This research paper delves into the application of the random forest algorithm for malware analysis, employing a numerical dataset consisting of a diverse range of malware attributes to train and appraise the model. We used the preprocessed dataset and have executed selection of feature to augment the performance of the model. Our findings reveal that the random forest algorithm attained exceptional accuracy in identifying malware samples. We also compared the efficacy of the random forest algorithm with other prevalent machine learning techniques deployed for malware analysis. Our results indicate that the random forest algorithm is a resilient and efficient technique for detecting malware and can be refined further by employing deep learning methodologies. This research has far-reaching implications for the development of automated malware analysis tools and can contribute substantially to the ongoing battle against malware threats.**

***Index Terms*—Malware analysis, Feature selection, Feature extraction, Decision trees, Machine learning.**

I. Introduction

The increasing prevalence and sophistication of malware pose a significant threat to the computer system security and network. Traditional signature-based antivirus software has proven to be insufficient in detecting new and evolving malware strains, leading to the development of more advanced techniques such as machine learning-based malware detection. Machine learning algorithms are a big help to analyze data is bulk amounts to learn patterns and identify malicious behavior, allowing for more data with accuracy and efficient recognition of malware.In this research paper, we expound upon a novel methodology aimed at bolstering the domain of malware detection through the application of machine learning, specifically honing in on the utilization of the distinguished random forest algorithm. Random forest is a type of ensemble learning method that combines multiple decision trees to improve the accuracy and robustness of the model. We use a publicly available dataset of malware samples to train and evaluate our model, with a focus on feature selection techniques and performance evaluation metrics. We undertake a rigorous juxtaposition of our random forest model vis-à-vis other prevalent machine learning paradigms utilized in the realm of malware detection. Conspicuously, this entails a comparative analysis of Support Vector Machines (SVM), Decision Trees, and Logistic Regression.Our objective is to contribute to the development of more effective and efficient malware detection systems by investigating the potential of machine learning-based techniques. To accomplish this goal, we evaluate the performance of our model on different types of malware, including viruses, worms, trojans, and other types of malware. We also consider the impact of various feature selection techniques and performance evaluation metrics on the accuracy and effectiveness of the model. An example of a machine learning system that is ca- pable of articulating the fundamental principles underlying the patterns it has identified is an interpretable machine learning model. Interpretable models are distinguished from traditional black-box models, such as deep neural networks, in that they can provide explicit explanations for their predictions. The transparency and accountability of machine learning systems can be enhanced by using interpretable models, which are particularly valuable in sensitive areas like healthcare where decisions made by machine learning systems can have significant impact on human lives [1]. Cyberattacks are a threat in the world of growing modern technology due to the increasing dependence on technology in our daily lives. The term ”cyberattack” refers to the malicious exploitation of system vulnerabilities for various purposes, such as unauthorized ac- cess, data theft, manipulation, or destruction. To protect against cyber-attacks, it is essential to implement robust cybersecurity measures, including the use of updated software, firewalls, and advanced security protocols. Additionally, providing regular training and education for users to recognize and avoid potential threats can also be highly effective in preventing cyber attacks [2]. The term ”malware” is a comprehensive term that covers a broad spectrum of dangers, including viruses, Wipers, trojan horses, scare ware, rogue software, ransom ware, spyware, adware and more. Malicious software denotes a form of software application that executes within a computer system devoid of user awareness or explicit consent. The definition of malware encompasses any code that is intentionally designed to harm a computer system or steal sensitive information. The utilization of malware has evolved into a progressively intricate and pervasive phenomenon, posing a substantial menace to the realm of computer security. It is of utmost importance for both individuals and entities to maintain a state of watchfulness

Identify applicable funding agency here. If none, delete this.

in their endeavours to shield themselves from the onslaught of malware assaults [3]. Machine learning algorithms have the capability to enhance their predictive performance by utilizing feedback on their past performance to refine their methods.These algorithms are trained using large datasets and rely on iterative processes to learn from patterns and develop predictive models. As these algorithms learn, they receive feedback on how well they have performed on previous tasks, and they incorporate this information to improve their predictive accuracy. Through this feedback loop, machine learning algorithms continuously refine their predictive capabilities by adjusting their internal parameters and optimizing their models [4]. The prevalence and complexity of modern malware represents a significant risk to the security of contemporary websites. Malware is a type of software designed to cause harm, disrupt system operations, and exploit vulnerabilities in computer systems. With the increasing sophistication of malware, modern web- sites are particularly susceptible to attacks, with malicious actors exploiting security loopholes to gain access to sensitive data and cause damage to critical systems. The threat posed by modern malware is a growing concern for businesses, organizations, and individuals, who must remain vigilant in their efforts to protect against cyberattacks and secure their online presence [5]. To identify behavioural similarities among members of the same malware family, both static and dynamic learning approaches can be utilized. Static learning refers to the analysis of the structure and characteristics of the malware's code without executing it.Conversely, dynamic learning entails the surveillance of malware conduct as it operates within a constrained setting. By using these two approaches, researchers and security professionals can identify the commonalities in the behaviour and characteristics of malware within the same family, which can help in detecting and mitigating potential threats. The use of these learning methods is essential in identifying and categorizing new and emerging malware strains, which is critical in developing effective countermeasures and protecting computer systems from potential attacks [6].

II. LITERATURE RIVIEW

Numerous investigations have delved into the utilization of machine learning algorithms for the purpose of identifying and detecting malware. In a study by Kolosnjaji et al. (2016), a comprehensive survey of machine learning techniques for malware detection was conducted, highlighting the advantages and limitations of various approaches. The study concluded that ensemble methods such as random forests and boosting algorithms were among the most effective techniques for malware detection. Another study by Xia et al. (2015) used a dataset consisting of over 7,000 malware samples and applied a variety of machine learning algorithms, including decision trees, support vector machines (SVMs), and neural networks. The study found that SVMs and neural networks achieved high detection rates, but random forests provided the best trade-off between accuracy and efficiency.Similarly, a study by Abraham et al. (2021) explored the performance of several machine learning algorithms, including random forest, decision tree, k-nearest neighbors, and SVM, on a dataset of Android malware samples. The study found that random forests achieved the highest accuracy and lowest false positive rate. These studies demonstrate the potential of machine learning algorithms for malware detection and highlight the effectiveness of random forest algorithm in particular.Nikam and Deshmuh (2022) evaluated the performance of different machine learning classifiers for detecting malware. The assessment encompassed a juxtaposition of the precision, recall, F1-score, and accuracy pertaining to a range of algorithms, including Random Forest (RF), Support Vector Machine (SVM), Naive Bayes (NB), k-Nearest Neighbor (k-NN) and Decision Tree (DT) algorithms. The experiments were conducted on a publicly available dataset, and the results showed that the RF classifier outperformed the others with an accuracy of 98.23 percent .Akhtar and Feng (2022) proposed an anomaly detection technique based on the Internet of Things (IoT) for mobile sensing. The authors used the IOTA (Internet of Things Analytics) platform to collect data from mobile sensors and then applied machine learning algorithms for anomaly detection. The proposed technique achieved an accuracy of 95 and showed better performance than the traditional methods.Sharma, Krishna, and Sahay (2017) used machine learning techniques for detecting advanced malware. They compared the performance of different classifiers, including SVM, RF, Logistic Regression (LR), and DT, using a collection of malignant and harmless programs as the dataset. The outcomes unveiled that the SVM classifier attained the utmost accuracy, reaching an impressive 98.7 percent. In 2015, Zhao, Zhang, Su, and Li introduced Fest, a tool designed to perform feature extraction and selection for the identification of Android malware. This application harnessed a hybrid approach, amalgamating both static and dynamic analysis methodologies to extract feature sets from the codebase. Subsequently, a feature selection mechanism was employed.

III. MATERIAL AND METHODOLOGIES

The dataset contained a total of 10,000 malware samples and 4,000 benign files, each labeled as either malicious or benign. The malware samples were collected from various sources and were preprocessed to extract a range of features, including static and dynamic attributes, such as API calls, file size. We employed the random forest algorithm, a type of ensemble learning method, for detecting malware in the dataset. The algorithm was trained using the preprocessed features extracted from the dataset. For improving the performance of the model, we applied feature selection techniques such as Re- cursive Feature Elimination (RFE) and Principal Component Analysis (PCA). We used a stratified 5-fold crossvalidation approach for evaluating the performance of the model, with thedataset being randomly divided into five equal parts, each used as a validation set once while the remaining four parts are usedas a training set. We measured the performance of the model using metrics such as recall, precision and accuracy and F1-score. In addition to random forest, we also evaluated the performance of other commonly used machine learning algorithms, including Support Vector Machines (SVM), Decision Tree, and Logistic Regression. The algorithms were trained and evaluated using the same dataset and cross-validation approach as used for random forest.

A. Models

Overall, the selected data points were carefully chosen to provide insights into the behavior of malware and potential indicators of malicious activity. By using these data points in our model, we aimed to better understand the nature and characteristics of malware, and develop effective techniques for detecting and mitigating it. In the testing phase, the algorithm evaluates each decision tree and computes the majority vote of their predictions using a process called bagging. This approach reduces the variance of the model and improves its accuracy.

- This method of harmonizing diverse decisions into a unified perspective functions as an effective variance mitigator, effectively ameliorating the propensity for wide-ranging deviations within the model's predictive capacity and concurrently amplifying its overall precision.
- By meticulously tempering and homogenizing these distinct and potentially discordant predictions, this approach obviates the pitfalls of exaggerated oscillations and deviations, consequently engendering an elevated echelon of precision and sagacity to the model's predictive acumen on a global scale.
- The model, epitomizing the amalgamation of intricate decision trees and the harmonious symphony of bagging, emerges as an exemplar of algorithmic virtuosity. Its dexterity in deciphering the enigmatic language of malware, forged through the fusion of myriad insights and the art of consensus-building, positions it as a sentinel against the ever-mutating choreography of digital threats.
- This multiplicity of decision trees transforms into topic wherein the culmination of individual voices engenders a robust and resilient final proclamation.

B. Dataset

- Within the context of my doctoral research, the provided dataset plays an indispensable role, serving as a crucial and fundamental component. My research endeavors are primarily centered upon harnessing the potential of Deep Learning techniques, with the overarching objective of effectively identifying and categorizing malware. To be more specific, this dataset consists of meticulous static analysis data that has been meticulously derived from the pe imports elements, which are integral parts of Cuckoo Sandbox reports. This carefully curated collection of static analysis data consists of the most significant and influential top 1000 imported functions.
- Additionally, instances of PE goodware were obtained from both portable apps.com and various directories associated with Windows 7 x86. By expending considerable efforts and ensuring the precise organization of this extensive dataset, I aspire to forge significant advancements in the domain of malware detection. These advancements will be made possible through the astute and intricate application of advanced computational models and techniques, thereby enriching the current landscape of malware identification and classification methodologies.

1) Features:

- Column name: hash
  Description: MD5 hash of the example
  Type: 32 bytes string
- Column name: GetProcAddress
  Description: Most imported function (1st)
  Type: 0 (Not imported) or 1 (Imported)
- Column name: LookupAccountSidW
  Description: Least imported function (1000th)
  Type: 0 (Not imported) or 1 (Imported)
- Column name: malware
  Description: Class
  Type: 0 (Goodware) or 1 (Malware)
- The dataset under examination in this research paper consists of 47,580 entries and 1,002 columns, with data types including int64 and object(1). The columns are labeled from 0 to 47579, and are associated with cyber security, as indicated by their names such as "Hash" and "Malware". The dataset is of considerable size, with a memory usage of approximately 370MB, Conveying intricate information and an extensive volume of data. Notably, the use of the int64 data type suggests extensive numerical data related to cyber security metrics, such as network traffic. The richness and complexity of this dataset make it an intriguing resource for researchers and practitioners in the field of cyber security, as it has the potential to provide valuable insights into various cyber threats and related metrics. This paper will explore the dataset in-depth, with the aim of uncovering novel findings and contributing to the broader understanding of cyber security.
- In our model, we focused on developing a malware detection model using a carefully selected dataset. To train our model, we first split the dataset into X labels and Y labels. The X-label included all 47,580 rows of data, with all columns except for the 'Malware' column. The Y-label, on the other hand, consisted of the 'Malware' column, with all corresponding entries.
- We then further split the X and Y labels into X train, Xtest, Y train, and Y test. This step was taken to ensure thatour model was trained and tested on independent and non-overlapping data. Particularly, we separated the dataset into 20training our model.
- By splitting our dataset in this way, we aimed to avoid over fitting and ensure that our model could generalize well to new and unseen data. The training data process of fitting our model to the data involved utilizing the

training data, whereas the evaluation of our model took place using the testing data.

- It's performance and estimate its ability to generalize to new.
- Overall, the process of splitting the dataset into X and Y labels, and further dividing it into training and testing sets, was a crucial step in the development of our malware detection model. By using this approach, we aimed to develop a robust and effective model for detecting malware, which could have practical applications in the field of cyber security. Fig.1 Showing the classification of the dataset into sub training and testing model.

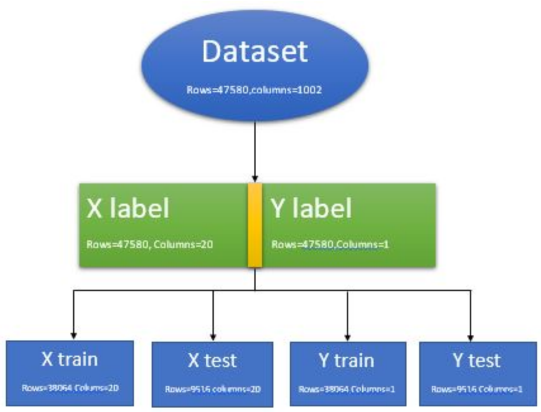

Fig. 1. Splitting of Data

- In our malware model, we carefully selected a subset of the dataset that we deemed to be the most crucial for our analysis. The following data points were included in our model: CloseHandle, GetProcAddress, ExitProcess, WriteFile, GetLastError, WinHttpOpen, Free library, Sleep, GetStdHandle, MultiByteToWideChar,bind, RegEnumKeyEx, WinHttpOpen, LookupAccountSidWe controlfp, WinExec, GetSecurityDescriptorDacl,FindFirstFreeAc, GetTimeFormatW, and malware.
- Each of these data points was chosen for its potential relevance to the behavior of malware. For example, GetProcAddress is a function that retrieve the address of an exported variable or function on or after a specified dynamic-link library (DLL), which could be used by malware to load additional functionality. Similarly, WriteFile is a function that writes data to a file or input/output (I/O) device, which could be used by malware to persistently store information or communicate with a command and control server. Overall, the selected data points were carefully chosen to provide insights into the behavior of malware and potential indicators of malicious activity. By using these data points in our model, we aimed to better understand the nature and characteristics of malware, and develop effective techniques for detecting and mitigating it.

Fig.2 Showing the Feature Selection upon which the model is based.

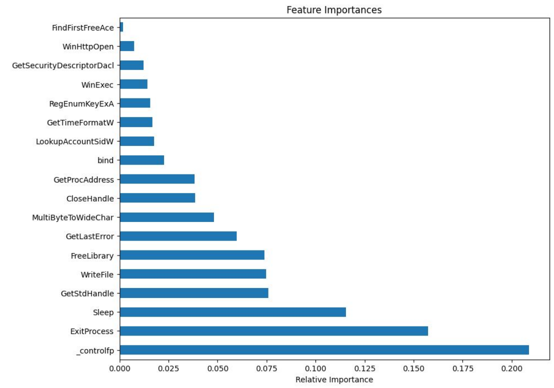

Fig. 2. Feature Selection

C. Equations

The gini index of the formula is used to determine how the nodes should bifurcate within the structure of a decision tree

$$Gini = 1 + \sum_{i=1}^{c}(\sum_{i=1}^{\infty} 1(\frac{8}{[(4n-3)(4n-1)]})) \quad (1)$$

The formula Eq (1) is used by the class and probablity for calculating the Gini index for every branch at a node involves assessing the likelihood of each branch occurring.

IV. PROPOSED ARCHITECTURE

The proposed architecture for the random forest model in our research paper includes the following steps: Collection of data: Collecting a large dataset of both malware and benign files is the first step of the analysis. This dataset is then preprocessed and labeled accordingly. Feature Selection and Extraction: Next, a set of features is extracted from the dataset to be used as input for the model. This can include file header information, file size, byte sequences, and other relevant metadata. These features are then analyzed and the most important ones are selected through a feature selection process. Fig.3 Showing the model classification

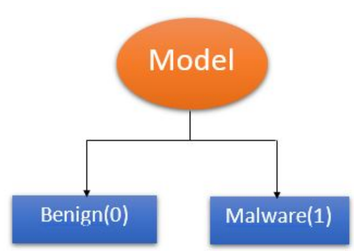

Fig. 3. FLowchart of model

1) Train-Test Split:: The preprocessed dataset is split into two distinct groups: a testing set and a training set. The testing set is employed to gauge the model's performance, whereas the training set is utilized for training the random forest classifier.

Training of classifier of Random Forest: The selected features and training set are then used to be trained the random forest classifier. During training of the model, the classifier builds a decision tree ensemble that is used to study and classify files as per malware or benign based on the selected features.

Trained Classifier Validation: The trained classifier is validated to ensure that it is accurately classifying the files. This validation is performed using techniques such as confusion matrix cross-validation, F1 score and precision. Fig.4 Showing the workflow of the model.

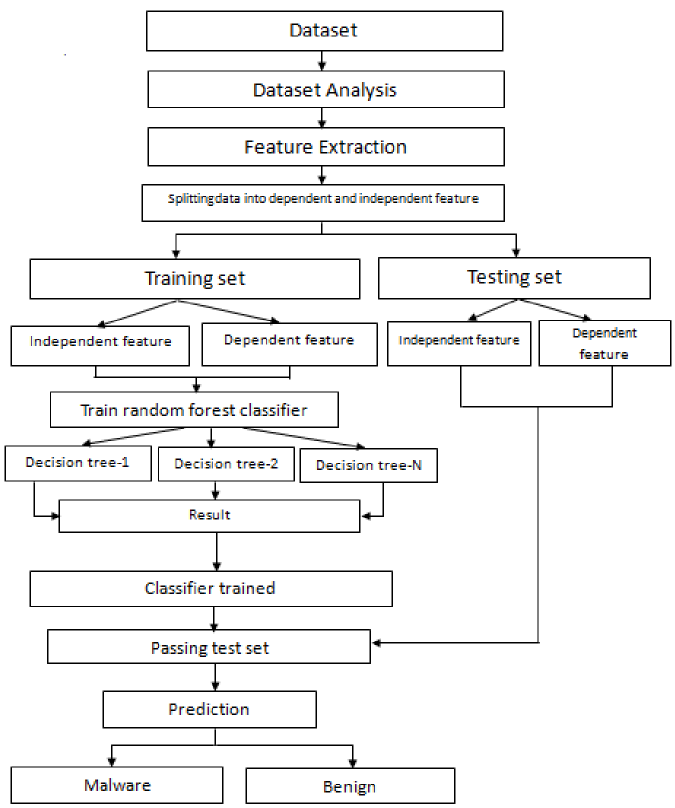

Fig. 4. Working Flowchart of the model

Predicting Malware or Benign: After the validation, the trained classifier can be used to predict whether a file is malware or benign. This is done by feeding the file's extracted features into the classifier and obtaining a prediction.

Overall, the proposed architecture for our malware analysis project using random forest model involves a series of steps that include data collection, feature extraction and selection, train-test split, training the random forest classifier, validation, and predicting malware or benign. By following this architecture, we can obtain an accurate and reliable malware classification model that can be used to identify and mitigate potential security threats

V. RESULT AND DISCUSSION

One of the predominant factors in machine learning model evaluation is its accuracy. important metrics used to evaluate its performance. In our research paper on malware analysis, the accuracy results are presented for both the testing phases and training of model. The train accuracy achieved was 0.9716792770071458, while the test accuracy was 0.9713114754098361.

Table I : Showing the objective and algorithm that is used to predict the accuracy of the model.

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP}$$

S.No	Description	Result
1	Objective	Detection of malware files
2	Algorithm used	Random forest classifier
3	True prediction	9365
4	False Prediction	151
5	Accuracy	97.65

TABLE I
ACCURACY PREDICTED BY THE MODEL

Fig.5 Representing the accuracy of the model

X-axis representing the number of estimators

Y-axis representing the accuracy

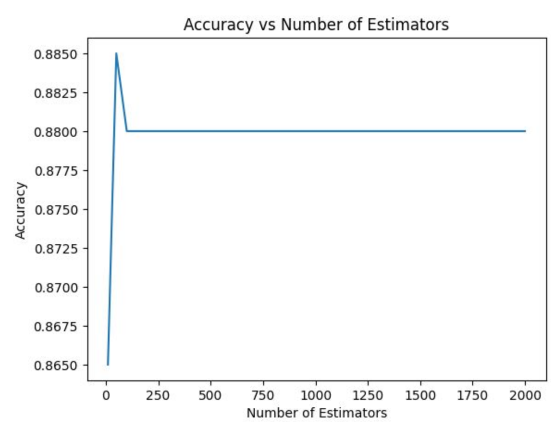

Fig. 5. Graph of Accuracy Vs Number of Estimators

Furthermore, the utilization of the confusion matrix was instrumental in assessing the performance metrics, including precision, recall, and F1 score, on the test set. The examination of the confusion matrix uncovered that the model accurately recognized 9100 instances of malware and 143 cases of benign samples. Notably, the model's precision stood at 0.9728458413512936, its recall at 0.9979164382059437, and its F1 score at 0.9852216748768473.

$$Precision = \int_{i=0}^{1} (\sum_{1}^{\infty} 1 \frac{-1^{n+1}}{2^{2n-1}} * sin(\frac{n\pi}{3})$$

$$\int_{0}^{1} sin^2x + cos^2x \int_{0}^{1} (\frac{1}{cos^2x}) + (\frac{tan^2x}{cos^2x})$$

$$w(\theta) * Precision(\theta) * dRecall(\theta)$$

S.No	Description	Result
1	Model Prediction	Malware and Benign
2	Accuracy	98.72
3	F1 score	98.72
4	Recall	99.60
5	Precision	97.98

TABLE II
SHOWING THE F1 SCORE, PRECISION AND RECALL OF THE MODEL

$$Recall = 1 - \frac{FN}{TP + FN}$$

Fig.6 Showing the Precision curve for the datasets provided to the model

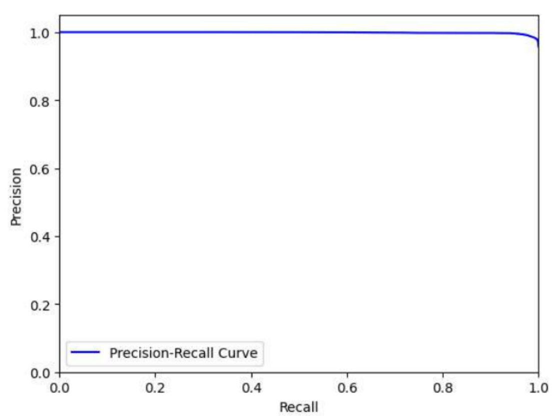

Fig. 6. Precision curve

$$F1Score = \frac{TP * 2}{TP * 2 + FP + FN}$$

CONFUSION MATRIX : The confusion matrix functions as a table-like instrument utilized to gauge the effectiveness of a procedure or approach of classification models. When it comes to test results, the confusion matrix shows that out of 9516 samples, 187 are classified as False Positive (false when they should be positive),36 are classified as False Negative (negative but expected to be positive), and 9083 are classified as true negative (predicted).value is negative and negative) and 210 true are classified as positive (price is predictable and good). Precision for this metric is 0.97284584135129 and Recall is 0. Fig.7 Showing the Precision curve for the gain and loss

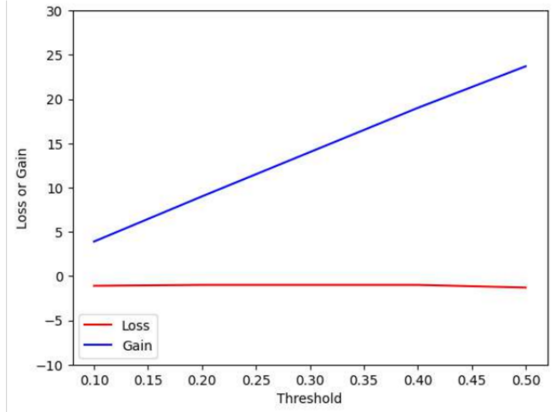

Fig. 7. Precision curve for gain and loss

Similarly, the confusion matrix for the Cross Validation Evaluation shows that the model correctly classified 36439 out of 36532 malware samples and 93 out of total of 1532 samples that were not classified as malware, the model made an error by classifying 483 of them as malware and 1049 actual malware samples as non-malware. Despite this, the model demonstrated impressive levels of accuracy, precision, recall, and F1 score across both the test set and the cross-validation set.

Our main goal is to contribute to continued efforts to improve the security of computer systems and networks. We do this by developing accurate and effective malware detection systems. We hope this research article will contribute to efforts to improve cybersecurity and prevent threats posed by malware. To achieve this, we present our findings in a systematic way. Within the segment dedicated to relevant research, we thoroughly examine prior efforts concerning the utilization of machine learning for the detection of malware. In the materials and methodology segment, we provide an account of the data employed in the experiment along with the approach taken for its selection. Within the Results and Discussion portion, we showcase our experimental outcomes and engage in an in-depth analysis of these results. Fig.8 Shows the Relation between the Predicted values and the actual values .

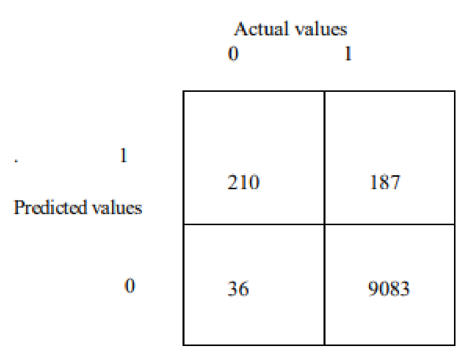

Fig. 8. Relation of Predicted values and Actual values

Finally, in the conclusion and future studies section, we present our findings and future research directions.

References

[1] Feng, T.; Akhtar, M.S.; Zhang, J. The future of artificial intelligence in cybersecurity: A comprehensive survey. EAI Endorsed Trans. Create. Tech. 2021, 8, 170285. [CrossRef]

[2] Nikam, U.V.; Deshmuh, V.M. Performance evaluation of machine learning classifiers in malware detection. In Proceedings of the 2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), Ballari, India, 23–24 April 2022; pp. 1–5. [CrossRef]

[3] Akhtar, M.S.; Feng, T. IOTA based anomaly detection machine learning in mobile sensing. EAI Endorsed Trans. Create. Tech. 2022, 9, 172814.

[4] Sharma, S.; Krishna, C.R.; Sahay, S.K. Detection of advanced malware by machine learning techniques. In Proceedings of the SoCTA 2017, Jhansi, India, 22–24 December 2017

[5] Zhao, K.; Zhang, D.; Su, X.; Li, W. Fest: A feature extraction and selection tool for android malware detection. In Proceedings of the 2015 IEEE Symposium on Computers and Communication (ISCC), Larnaca, Cyprus, 6–9 July 2015; pp. 714–720.

[6] Gibert, D.; Mateu, C.; Planes, J.; Vicens, R. Using convolutional neural networks for classification of malware represented as images. J. Comput. Virol. Hacking Tech. 2019, 15, 15–28. [CrossRef]

[7]] Gandotra, E., Bansal, D., and Sofat, S, Malware Analysis and Classification: A Survey. Journal of Information Security, Volume 5, pp. 56-64. doi: 10.4236/jis.2014.52006.

[8]] F. Rosenblatt, “The perceptron: a probabilistic model for information storage and organization in the brain,” pp. 89–114, 1988.

[9]] N. Cristianini and J. Shawe-Taylor, An introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press, March 2000

[10]] P. Baldi, S. Brunak, Y. Chauvin, C. A. Andersen, and H. Nielsen, “Assessing the accuracy of prediction algorithms for classification,” Bioinformatics, no. 5, pp. 412–424, May 2000.

[11] C. Cortes and V. Vapnik, “Support-vector networks,” in Machine Learning, 1995, pp. 273–297.

[12]] M. Stamp, S. Attaluri, and S. McGhee, “Profile hidden markov models and metamorphic virus detection,” Journal in Computer Virology, 2008.

[13] K. Rieck, T. Holz, C. Willems, P. D¨ussel, and P. Laskov, “Learning and classification of malware behavior,” in DIMVA ’08: Proceedings of the 5th international conference on Detection of Intrusions and Malware, and Vulnerability Assessment. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 108–125

[14]] V. N. Vapnik, The Nature of Statistical Learning Theory (Information Science and Statistics). Springer, November 1999

[15]] Y. Freund and R. E. Schapire, “Large margin classification using the perceptron algorithm,” in Machine Learning, vol. 37, 1999, pp. 277–296.

[16]] Y. Ye, D. Wang, T. Li, and D. Ye, “Imds: intelligent malware detection system,” in KDD, P. Berkhin, R. Caruana, and X. Wu, Eds. ACM, 2007, pp. 1043–1047.

[17] E. Konstantinou, “Metamorphic virus: Analysis and detection,” 2008, Technical Report RHUL-MA-2008-2, Search Security Award M.Sc. thesis, 93 pages.

[18] M. R. Chouchane, A. Walenstein, and A. Lakhotia, “Using Markov Chains to filter machine-morphed variants of malicious programs,” in Malicious and Unwanted Software, 2008. MALWARE 2008. 3rd International Conference on, 2008, pp. 77–84

[19]] T. Mitchell, Machine Learning. McGraw-Hill Education (ISE Editions), October 1997.

[20] F. Rosenblatt, “The perceptron: a probabilistic model for information storage and organization in the brain,” pp. 89–114, 1988.